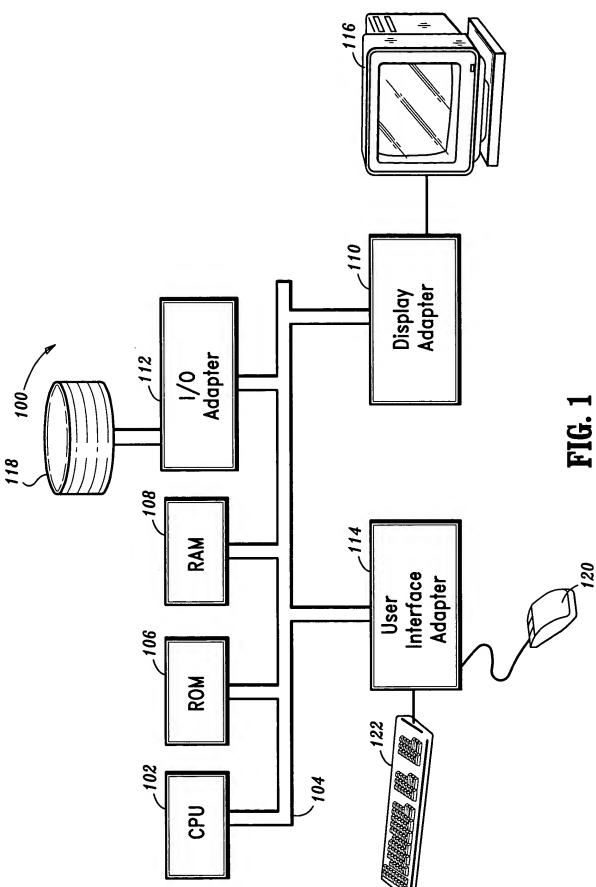


1/6 Burke et al. YOR9-2000-0038US2 (CPA) (8728-348)





2/6 YOR9-2000-0038US2 (8728-348) Start 210 Identify methods for which interprocedural optimization can be applied and specialized versions can be created, based upon predefined critéria 212 Generate optimized code for the identified methods, based upon parametric data flow analysis 214 Generate extant safety test 216 Execute extant safety test during runtime 218 Did No Yes the extant safety test pass? 222 220 Invoke the optimized code, in Invoke the identified methods place of the identified methods

End

FIG. 2



3/6 YOR9-2000-0038US2 (8728-348)

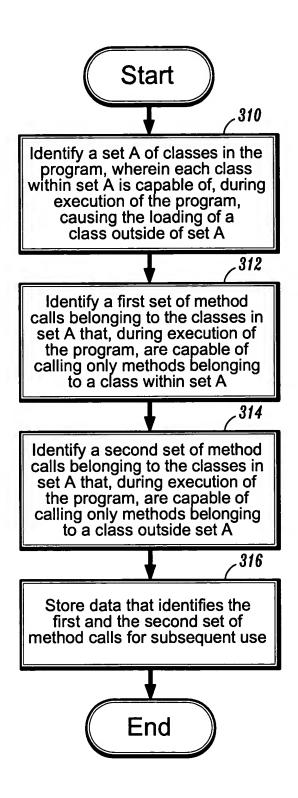


FIG. 3



4/6 YOR9-2000-0038US2 (8728-348)

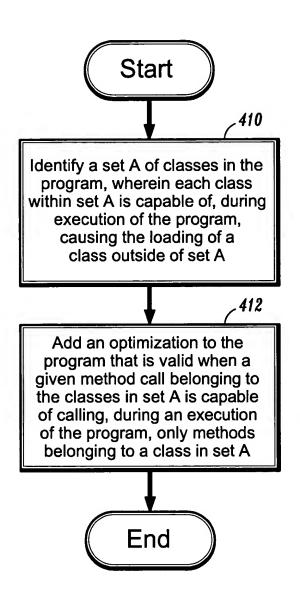


FIG. 4



5/6 YOR9-2000-0038US2 (8728-348)

Start

.510

Identify a set A of classes in the program, wherein each class within set A is capable of, during execution of the program, causing the loading of a class outside of set A

.510a

Add a first predefined bit to the class table of each class corresponding to set A

510b

Add a second predefined bit to the class table of each class not corresponding to set A

.512

Identify a set B of method calls belonging to the classes in set A that, during execution of the program, are capable of calling only methods belonging to a class outside set A

_514

For a given method call in the set B, generate optimized code for replacing the given method call

.516

For the given method call in the set B, generate test code that triggers execution of the optimized code when the given method call calls, during an execution of the program, only methods belonging to a class within the set A; the test code is generated to query the class table corresponding to the run time class of a reference variable associated with the given method call to determine whether the bit corresponds to the first or the second predefined value

FIG. 5

End



6/6 YOR9-2000-0038US2 (8728-348)

